(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
3 January 2002 (03.01.2002)

PCT

(10) International Publication Number
WO 02/01364 A2

(51) International Patent Classification[7]: G06F 12/00

(21) International Application Number: PCT/US01/17845

(22) International Filing Date: 1 June 2001 (01.06.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/602,011 23 June 2000 (23.06.2000) US

(71) Applicant *(for all designated States except US)*: **INTEL CORPORATION** [US/US]; 2200 Mission College Boulevard, Santa Clara, CA 95052 (US).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **COULSON, Richard** [US/US]; 17454 NW Gilbert Lane, Portland, OR 97229 (US).

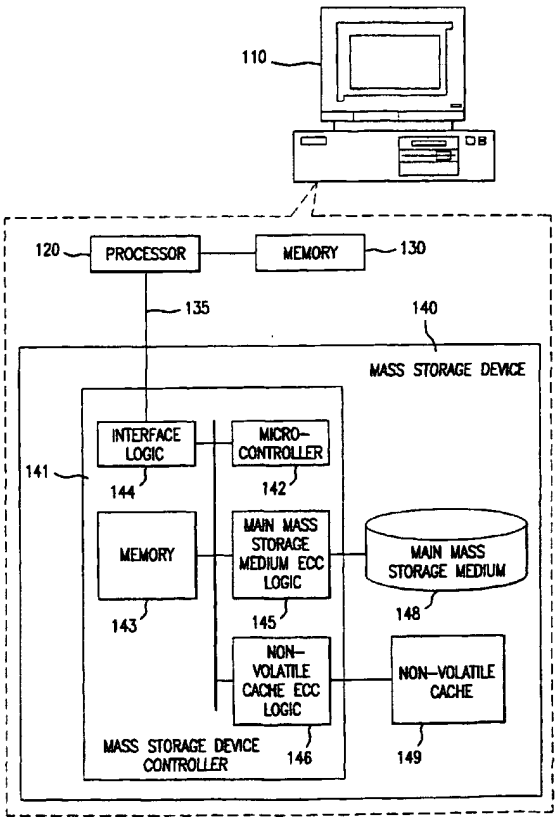
(74) Agents: **HAILS, Robert, L., Jr.** et al.; Kenyon & Kenyon, Suite 600, 333 West San Carlos Street, San Jose, CA 95110 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE,

[Continued on next page]

(54) Title: NON-VOLATILE CACHE INTEGRATED WITH MASS STORAGE DEVICE



(57) Abstract: Apparatus and methods relating to a non-volatile mass storage device including a non-volatile cache.

WO 02/01364 A2

IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

NON-VOLATILE CACHE INTEGRATED WITH MASS STORAGE DEVICE

FIELD OF THE INVENTION

Embodiments of the present invention relate to data storage. More particularly, embodiments of the present invention relate to a mass storage device having a non-volatile cache.

BACKGROUND OF THE INVENTION

A computer can store data both in a volatile memory and a non-volatile mass storage device. Examples of volatile memory include, but are not limited to, dynamic random access memory (DRAM), static random access memory (SRAM), Rambus dynamic random access memory (RDRAM), etc. Examples of a non-volatile mass storage device include, but are not limited to, a hard disk drive, a 3.5-inch diskette, a 5.25-inch floppy diskette, a ZIP® disk (e.g., manufactured by Iomega Corporation of Roy, Utah), a Jaz® disk (e.g., manufactured by Iomega Corporation), an LS-120 Superdisk (e.g., manufactured by Imation Corporation of Oakdale, Minnesota), a rewritable Digital Versatile Disc (DVD-RAM), a Read/Write Compact Disk (CD-RW), a magnetic mass storage device, an optical mass storage device, a magneto-optical mass storage device, a holographic storage device, etc. Examples of data stored in a non-volatile memory of a computer include computer instructions (e.g., an operating system, one or application programs, etc.) and data that is accessed by computer instructions.

When a volatile memory loses power, the data stored on the volatile memory is typically lost. Non-volatile mass storage devices, such as hard disk drives, typically do

not lose the data stored thereon when the non-volatile mass storage device loses power (e.g., when power to the computer is turned off, due to a power outage, etc.). Non-volatile memory, however, generally has significantly greater access times for retrieval and storage of data as compared to volatile memory. Non-volatile memory is also typically less expensive per storage unit (e.g., per megabyte, per gigabyte, etc.) than certain volatile memory such as DRAM, SRAM, RDRAM, etc.

In addition to volatile memory and non-volatile memory, a computer typically includes a processor that can perform operations based on instructions and data. Instructions and data to be operated on by the processor can be copied from the slower, non-volatile mass storage device (e.g., a hard disk drive, etc.) to the faster, volatile memory (e.g., a DRAM main memory, an SRAM cache, etc.) because the faster, volatile memory typically has a significantly lesser memory access time than the non-volatile mass storage device. Processor performance and computer performance can be enhanced when memory access times are reduced.

Because the memory access time for a non-volatile mass storage device (e.g., disk drive) is generally greater than the memory access time for the volatile memory (e.g., DRAM main memory, SRAM cache, etc.), the non-volatile mass storage device is often a performance bottleneck. Known disk drives include a volatile cache (e.g., a DRAM cache, an SRAM cache), but such volatile caches are typically part of the disk drive's microcontoller's main memory address space and thereby byte addressed. In view of the foregoing, it can be appreciated that a substantial need exists for methods and apparatus which can enhance computer system performance.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows an illustration a computer system in accordance with an embodiment of the present invention.

FIG. 2 shows an illustration of the organization of a non-volatile cache in accordance with an embodiment of the present invention.

FIG. 3 shows a method in accordance with an embodiment of the present invention.

FIG. 4 shows a method in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION

According to an embodiment of the present invention, a non-volatile mass storage device includes a non-volatile cache. The non-volatile cache can store information that is written to, or can be read from, the main mass storage medium of the non-volatile mass storage device. The non-volatile cache can have a lesser access time with respect to a data read as compared to the access time of a data read from the main mass storage medium of the non-volatile mass storage device. A computer system incorporating an embodiment of the present invention can have increased system performance when data stored on the non-volatile mass storage device is read from the non-volatile cache instead of from the main mass storage medium of the non-volatile mass storage device.

FIG. 1 shows an illustration a computer system ("computer") in accordance with an embodiment of the present invention. Computer 110 can include a processor 120 coupled to a memory 130. The term "coupled" encompasses a direct connection, an indirect connection, a direct communication, an indirect communication, etc. Processor

120 can be, for example, a Pentium® III processor manufactured by Intel Corporation of Santa Clara, California, an application specific integrated circuit (ASIC), a microcontroller, etc. Memory 130 encompasses devices that store digital information such as DRAM, RDRAM, SRAM, read only memory (ROM), flash memory, etc. In another embodiment, a system bus can provide a communication path between the processor 120 and system components. The system bus can be a Peripheral Component Interconnect (PCI) bus, an Extended Industry Standard Architecture Bus (EISA), etc.

In an embodiment, a chipset can be coupled to, and manages interaction between, the processor 120 and other system components, such as the memory 130, mass storage device 140, peripheral components attached to an expansion bus, etc. The term chipset encompasses a group of one or more integrated circuit chips that acts as a hub (or core) to transfer data between the processor and other system components. Examples of a chipset include the 820 and 810E chipsets made by Intel Corporation. A chipset can be a single integrated circuit chip or can comprise two or more integrated circuit chips. A chipset can include a memory control hub. In an embodiment including a PCI bus, a memory control hub can perform functionality known as "northbridge functionality." A chipset can include an input/output controller hub. In an embodiment including a PCI bus, an input/output controller hub can perform functionality known as "southbridge functionality."

Processor 120 is also coupled to mass storage device 140 via a communication path 135. In an embodiment, communication path 135 can be an Integrated Device Electronics (IDE) bus, an Enhanced IDE (EIDE) bus, an AT Attachment (ATA) bus, etc. In a further embodiment, the communication path 135 can be an expansion bus such as a

small computer system interface (SCSI) bus, an IEEE 1394 bus, a Universal Serial Bus (USB), etc.

Examples of mass storage device 140 include a hard disk drive, a ZIP® drive, a Jaz® drive, a CD-RW drive, a DVD-RAM drive, an LS-120 Superdisk drive, a magnetic storage device, an optical storage device, a magneto-optical storage device, a holographic storage device, etc. An example of a hard disk drive is the Cheetah 18XL hard disk drive manufactured by Seagate Technology Inc. of Scotts Valley, California.

Mass storage device 140 can include a mass storage device controller 141, a main mass storage medium 148, and a non-volatile cache 149. The mass storage device controller 141 can include a microcontroller 142, a memory 143, interface logic 144, main mass storage medium ECC (error control and checking) logic 145, and non-volatile cache ECC logic 146. In another embodiment of the present invention, mass storage device controller 141 can include one or more ASICs implementing functions to operate mass storage device 140 (e.g., a first ASIC incorporating a processor, a memory and interface logic and a second ASIC implementing ECC logic, etc.).

Microcontroller 142 can execute instructions stored in memory 143 to control the operations of mass storage device 140 (e.g., memory 143 can include read only memory that stores program instructions to initialize and operate mass storage device 140 upon power up, memory 143 can store instructions to perform reads of data from mass storage device 140, memory 143 can store instructions to control formatting of and/or writing data to mass storage device 140, etc.). The memory 143 can also include a buffer (e.g., a volatile memory buffer, etc.) that can temporarily hold data that is being written to, or read from, mass storage device 140.

Microcontroller 142 and interface logic 144 can process requests from processor 120 to read data from, or write data to, mass storage device 140. In an embodiment, mass storage device 140 is a hard disk drive and interface logic 144 can be IDE interface logic, EIDE interface logic, ATA interface logic, SCSI interface logic, Fibre Channel interface logic, InfiniBand interface logic, etc. In another embodiment, mass storage device 140 is a CD-RW drive or a DVD-RAM drive and interface logic 144 is ATA Packet Interface (ATAPI) logic. The term "logic" encompasses hardware, firmware, software, a combination thereof, etc.

In an embodiment, the mass storage device 140 is a hard disk drive, and the main mass storage medium 148 includes one or more platters. Each platter can have one or more recordable surfaces, and each recordable surface can be read/written by a particular read/write head. Each surface can be divided into a plurality of tracks; and each track can be divided into a plurality of physical sectors. When a hard disk drive has a plurality of recordable surfaces, the collection of all the tracks at the same radial location on all recordable surfaces is called a cylinder. The location of each physical sector of a hard disk drive can be specified by a physical address specifying the cylinder, head (surface), and physical sector.

In an embodiment, when a processor requests data from a hard drive, the request does not specify the physical address of the data but rather the logical address of the data. Logical addresses of hard disk data can include logical sector addresses, logical block addresses, etc. When a hard disk drive is formatted, the hard disk drive controller of the hard disk drive can organize all of physical sectors into logical sectors (e.g., assigning each physical sector a logical sector address, etc.), and a mapping algorithm can map each logical sector address to a particular physical address. When read/write requests to a hard

disk drive specify a logical sector address, the hard disk drive can be considered sector-oriented (i.e., sector addressable). A hard disk drive controller can also map collections of physical sectors into particular logical block addresses. When read/write requests to a hard disk drive specify a logical block address, the hard disk drive can be considered block-oriented (i.e., block addressable).

Non-volatile cache 149 can cache data that is typically stored on main mass storage medium 148. Non-volatile cache 149 can cache data read from, and written to, main mass storage medium 148. Non-volatile cache 149 can have a faster access time as compared to the access time of main mass storage medium 148. Accordingly, data reads from, and data writes to, mass storage device 140 can be completed faster as compared to data read/writes to main mass storage medium 148. In an embodiment, data read/writes to non-volatile cache 149 require less power as compared to data read/writes. Write-back algorithms (i.e., write backs from the non-volatile cache 149 to the main mass storage medium 148) can be implemented to increase cache performance. In another embodiment, write-through caching is implemented.

Non-volatile cache 149 can store a duplicate copy of frequently used mass storage device data and provide reduced access times for that frequently used data. In an embodiment, when data is read from main mass storage medium 148 and sent over communications path 135, a copy of the data can be stored in non-volatile cache 149. When data is sent to mass storage device 140 over communications path 135 to be stored, the data can be written to non-volatile cache 149 and written to main mass storage medium 148 according to a write-back algorithm, a write-through protocol, etc. Because the non-volatile cache 149 is a non-volatile memory, the data saved in the non-volatile cache is typically not lost when the power (e.g., power to the mass storage device 140, power to the

computer 110, etc.) is turned off. Non-volatile cache 149 can be any type of memory that can be read from/written to and retain its data when power external to the non-volatile cache 149 is removed or turned off.

Non-volatile cache 149 can be, for example, a flash memory, a battery backed-up DRAM, a battery backed-up SRAM, an atomic force probe memory, a magnetic RAM, a ferro-electric RAM, a holographic memory, a storage array, etc. In an embodiment, reads from non-volatile cache 149 can be destructive reads. Non-volatile cache 149 can be a block-oriented cache where data is accessed (e.g., written, read, etc.) in logical blocks. For example, when mass storage device 140 receives a data request (e.g., a data request to read data from mass storage device 140 onto communications path 135, a data request to write data to the mass storage device 140 from communications path 135, etc.), the data request can contain an address that is a logical block address, a logical sector address, etc. In an embodiment of the present invention, each of the data entries stored in non-volatile cache 149 corresponds to data stored at a logical address of main mass storage medium 148.

In an embodiment in which mass storage device 140 is a hard disk drive, a non-volatile cache 149 can increase system perform because data reads/writes from the non-volatile cache 149 do not require the warm-up time incidental in writing/reading data from hard disk drive platters (e.g., spinning the platters up to operational speed, positioning read/write heads, etc.). For example, the time required for system boot-up of computer 110 can be reduced when boot-up information (e.g., operating system data, device drivers, application data, etc.) is read from a non-volatile cache 149 as opposed to hard disk drive platters. Keystroke/mouseclick latency experienced by a user of computer 110 can also be reduced.

In an embodiment, cache management instructions can manage data reads from/writes to non-volatile cache 149. The cache management instructions can be stored in memory 143, can be part of a mass storage device driver, etc. Cache management instructions can make decisions regarding what data can be cached, what data can be evicted from the cache, what data can be written back to the main mass storage medium, when data write-backs occur, etc. Cache management instructions can also determine what data can be pre-fetched into the non-volatile cache 149. Cache management determinations, such as whether certain data should be cached in non-volatile cache 149, can be made using known cache management algorithms. For example, when a cache management algorithm determines that there is a low likelihood that the data read from mass storage device 140 will be read again in the near future (e.g., the data is part of an MP3 audio file, part of a WAV file, part of an AVI file, part of a streaming data file, etc.), then the data need not be cached. In addition, when it becomes advantageous to write data from non-volatile cache 149 to main mass storage medium 148, cache management instructions can determine which data to write back using, for example, a least recently used (LRU) algorithm, a random replacement algorithm, etc.

In an embodiment, a mass storage device driver for the mass storage device 140 can be a typical mass storage device driver (e.g., ATAPI.SYS in a WIN98 environment, etc.), and the cache management instructions can be stored in memory 143. In such an embodiment, the existence of non-volatile cache can be transparent to the operating system. In a further embodiment, the cache management instructions can be part of cache management logic of mass storage device controller 141.

Data read from main mass storage medium 148 can be processed by main mass storage medium ECC logic 145, and data read from non-volatile cache 149 can be

processed by non-volatile cache ECC logic 146. Each of main mass storage medium ECC logic 145 and non-volatile cache ECC logic 146 can detect errors at the bit or multi-bit level and correct the errors as the data is being sent to communications path 135.

FIG. 2 shows an illustration of the organization of a non-volatile cache in accordance with an embodiment of the present invention. A non-volatile cache 200 (e.g., non-volatile cache 149 of FIG.1) may store a plurality of cache entries 205. Each cache entry 205 can include a table entry field 210 and a data entry 220. Data entry 220 can include a valid field 222, a modified field 224, an address field 226, a data field 228, and an ECC field 229.

The table entry field 210 of a cache entry 205 can correspond to tag/index information for each data entry 220 of the cache entry 205. The plurality of table entry fields 210 can comprise a cache directory table 215, and the cache directory can be accessed to determine whether a specific set of data of a mass storage device (e.g., a disk sector of a hard disk drive, a logical block of a hard disk drive, etc.) is present in the non-volatile cache. In an embodiment, the table entry field 210 of a cache entry 205 can store an address (e.g., a logical sector address, a logic block address, etc.) of the data entry 220 of the cache entry 205. To determine whether a specific set of data of a mass storage device is present in a non-volatile cache, the cache directory table 215 can be searched using, for example, a known search algorithm. Alternatively, the cache directory can be sorted using a hashing algorithm. In an embodiment where the mass storage device is a hard disk drive, the presence of a requested disk sector in a non-volatile cache can be confirmed by comparing the sector address of the requested disk against the sector addresses stored in the cache directory table 215. In an embodiment, a non-volatile cache

can be a fully associative cache. In another embodiment, a non-volatile cache can be a set associative cache.

In an embodiment, each of the cache entries 205 stored in a non-volatile cache of a mass storage device corresponds to a logical block of a mass storage device. In another embodiment, each of the cache entries 205 stored in a non-volatile cache of a mass storage device corresponds to a sector (e.g., a physical sector, a logical sector) of a mass storage device.

In an embodiment that has 2,000,000 cache entries 205 in a non-volatile cache, there are 2,000,000 table entries in the cache directory table 215. In this embodiment, each table entry of the cache directory table 215 can be four bytes long, and the cache directory table 215 can use eight megabytes of non-volatile memory. The size of a cache directory table 215 can be reduced (and thus the speed of the average cache access increased) by including a block of multiple disk sectors in each cache entry.

In an embodiment, a data entry 220 of a cache entry 205 may have a valid field 222, a modified field 224, an address field 226, a data field 228, and an ECC field 229. Valid field 222 may be set to "valid" (e.g., one of a logical one and a logical zero, etc.) when cache entry 205 contains valid data and may be set to "invalid" (e.g., the other of a logical one and a logical zero, etc.) when cache entry 205 does not contain valid data. For example, when data is evicted from the non-volatile cache to allow other data to be cached in the non-volatile cache, the valid fields of the evicted data entries can be changed from "valid" to "invalid" to signify that such cache entries no longer store cached data. Modified field 224 can be set to "modified" if the data in cache entry 205 is not identical to the data stored in the corresponding location on the main mass storage medium of the mass storage device. For example, and referring to FIGS. 1 and 2, when data

corresponding to a logical address of the mass storage device 140 is written back to the mass storage device 140 and stored in the non-volatile cache 149, and that data was modified after having been read from the main mass storage medium 140, then the modified field 224 for the non-volatile cache entry 205 can be set to "modified" to indicate that the copy of the data corresponding to that logical address stored in the non-volatile cache 149 is different than the data corresponding to that logical address stored on the main mass storage medium 148. Modified field 224 can be referred to as a "dirty bit" and can be cleared when the data is successfully written-back to the main mass storage medium 148 and retained in the non-volatile cache 149.

Address field 226 can contain the address (e.g., logical address, physical address) for the data stored in the cache entry 205. In an embodiment in which the mass storage device is a hard disk drive, the address field 226 can store the sector address and can be referred to as a "sector identifier." In such an embodiment where logical blocks of data are cached in the non-volatile cache (e.g., each logical block corresponding to a plurality of sectors), address field 226 can contain the starting sector address of the logical block where each logical block has a known fixed size.

Data field 228 can store the data of cache entry 205. In an embodiment in which the mass storage device is a hard disk drive, data field 228 can contain a disk sector of data (e.g., 512 bytes), a logical block of data, etc. ECC field 229 can stores error correcting codes for the data of the cache entry 205. In an embodiment, each logical block of data is associated with an error correcting code.

FIG. 3 shows a method in accordance with an embodiment of the present invention. In an embodiment of the present invention, a mass storage device receives a request for data (e.g., a request from processor 120, a request from an operating system of

computer 110, a request from an application being executed on computer 110, etc.) (box 310). Whether the requested data is stored in a non-volatile cache of the mass storage device is determined (box 320). In one embodiment, the request for data includes an address, and that address is used to generate a cache hit or cache miss indication. In another embodiment, a cache directory includes addresses of the data cached in the non-volatile cache, and an indication that the requested data is cached in the non-volatile cache can be generated based on the cache directory and the address of the requested data. When the requested data is stored in the non-volatile cache, the data is read from the non-volatile cache (box 330), error checking and correction can be performed upon the read data (box 335), and the data can be sent from the mass storage device (box 340). When the requested data is not stored in the non-volatile cache, the data can be read from a main mass storage medium of the mass storage device ((box 360), error checking and correction can be performed upon the read data (box 365), the data can be sent from the mass storage device (box 370), and the data can be written to the non-volatile cache (box 375).

FIG. 4 shows a method in accordance with an embodiment of the present invention. A request to store data in non-volatile mass storage device is received (box 410). For example, a user editing a document in a word processing program can instruct the word processing program to save the document. As another example, the word processing program can include an auto save feature that causes the document to be saved periodically. As another example, an operating system can instruct that data stored in main memory (e.g., DRAM) be stored to a hard disk drive or other non-volatile memory (e.g., to free up room in the main memory for other data, etc.). A determination can be made whether the data is to be stored in a non-volatile cache (box 420). When a determination is made to store the data in the non-volatile cache, the data can be stored in

the non-volatile cache (box 430). In an embodiment, storing data in the non-volatile cache includes updating a cache directory table of the non-volatile cache (box 440). When a determination is made not to store the data in the non-volatile cache, the data can be stored in the main mass storage medium of the non-volatile mass storage device (box 450).

In accordance with an embodiment of the present invention, instructions adapted to be executed by a processor to perform a method are stored on a computer-readable medium. The computer-readable medium can be a device that stores digital information. For example, a computer-readable medium includes a ROM as is known in the art for storing software and/or firmware (e.g., microcode). The computer-readable medium can be accessed by a processor suitable for executing instructions adapted to be executed. The term "adapted to be executed" is meant to encompass any instructions that are ready to be executed in their present form (e.g., machine code) by a processor, or require further manipulation (e.g., compilation, decryption, or provided with an access code, etc.) to be ready to be executed by a processor.

Methods and apparatus in accordance with embodiments of the present invention can advantageously cache data read from, or written to, a mass storage device in a non-volatile cache. The non-volatile cache can retain its data even when external power to the non-volatile cache and the mass storage device is turned off. Embodiments of the present invention can reduce the access time for data read from, or written to, a mass storage device and thereby increase system performance of a system having a mass storage device.

Embodiments of methods and apparatus to store data in a non-volatile cache of a mass storage device have been described. In the foregoing description, for purposes of explanation, numerous specific details are set forth to provide a thorough understanding of the present invention. It will be appreciated, however, by one skilled in the art that the

present invention may be practiced without these specific details. In other instances, structures and devices are shown in block diagram form. Furthermore, one skilled in the art can readily appreciate that the specific sequences in which methods are presented and performed are illustrative and it is contemplated that the sequences can be varied and still remain within the spirit and scope of the present invention.

In the foregoing detailed description, apparatus and methods in accordance with embodiments of the present invention have been described with reference to specific exemplary embodiments. Accordingly, the present specification and figures are to be regarded as illustrative rather than restrictive.

WHAT IS CLAIMED IS:

1. An apparatus to store data, the apparatus comprising a non-volatile mass storage device including a non-volatile cache.

2. The apparatus of claim 1, wherein the non-volatile mass storage device is block-oriented and the non-volatile cache is block-oriented.

3. The apparatus of claim 1, wherein the non-volatile mass storage device is sector-oriented and the non-volatile cache is sector-oriented.

4. The apparatus of claim 1, wherein the non-volatile mass storage device is a hard disk drive.

5. The apparatus of claim 1, wherein the non-volatile mass storage device includes a main mass storage medium having a first non-volatile storage medium, the non-volatile cache includes a second non-volatile storage medium, the first non-volatile storage medium being a different type of non-volatile storage medium than the second non-volatile storage medium.

6. The apparatus of claim 5, wherein the main mass storage medium includes a platter having a recordable surface to store data.

7. The apparatus of claim 5, further comprising:

first error correction logic to process data read from the main mass storage medium; and

second error correction logic to process data read from the non-volatile cache.

8. The apparatus of claim 1, wherein the non-volatile cache includes a non-volatile storage array.

9. A computer system comprising:

a processor;

a memory coupled to the processor; and

a non-volatile mass storage device coupled to the processor, the non-volatile mass storage device having

a main mass storage medium, and

a non-volatile cache coupled to the main mass storage medium..

10. The computer system of claim 9, wherein the main mass storage medium includes a first non-volatile storage medium, the non-volatile cache includes a second non-volatile storage medium, the first non-volatile storage medium being a different type of non-volatile storage medium than the second non-volatile storage medium.

11. The computer system of claim 9, wherein the non-volatile mass storage device is block-oriented and the non-volatile cache is block-oriented.

12. The computer system of claim 9, wherein the non-volatile mass storage device is sector-oriented and the non-volatile cache is sector-oriented.

13. The computer system of claim 9, wherein the first non-volatile storage medium is a recordable medium selected from the group consisting of a magnetic recordable medium, an optical recordable medium, and a magneto-optical recordable medium.

14. The computer system of claim 9, wherein the second non-volatile storage medium is a storage medium selected from the group consisting of battery-powered dynamic random access memory, battery-powered static random access memory, flash memory, atomic probe storage memory, ferro-electric memory, and holographic memory.

15. A method of processing a request for data, the method comprising:

receiving a request for a first set of data stored in a non-volatile mass storage device;

determining that the first set of data is stored in a non-volatile cache of the non-volatile mass storage device;

reading the first set of data from the non-volatile cache; and

sending the first set of data.

16. The method of claim 15, wherein:

reading the first set of data from the non-volatile cache includes performing error correction on a subset of the first set of data; and

sending the data read from the non-volatile cache includes sending the error corrected subset of the first set of data.

17. The method of claim 16, further comprising:

receiving a request for a second set of data stored in the non-volatile mass storage device;

determining that the second set of data is not stored in a non-volatile cache of the non-volatile mass storage device;

reading the second set of data from a main mass storage medium of the non-volatile mass storage device;

writing the second set of data to the non-volatile cache; and

sending the second set of data.

18. The method of claim 15, wherein the request for a first set of data is a block-oriented request.

19. The method of claim 15, wherein the request for a first set of data is a sector-oriented request.

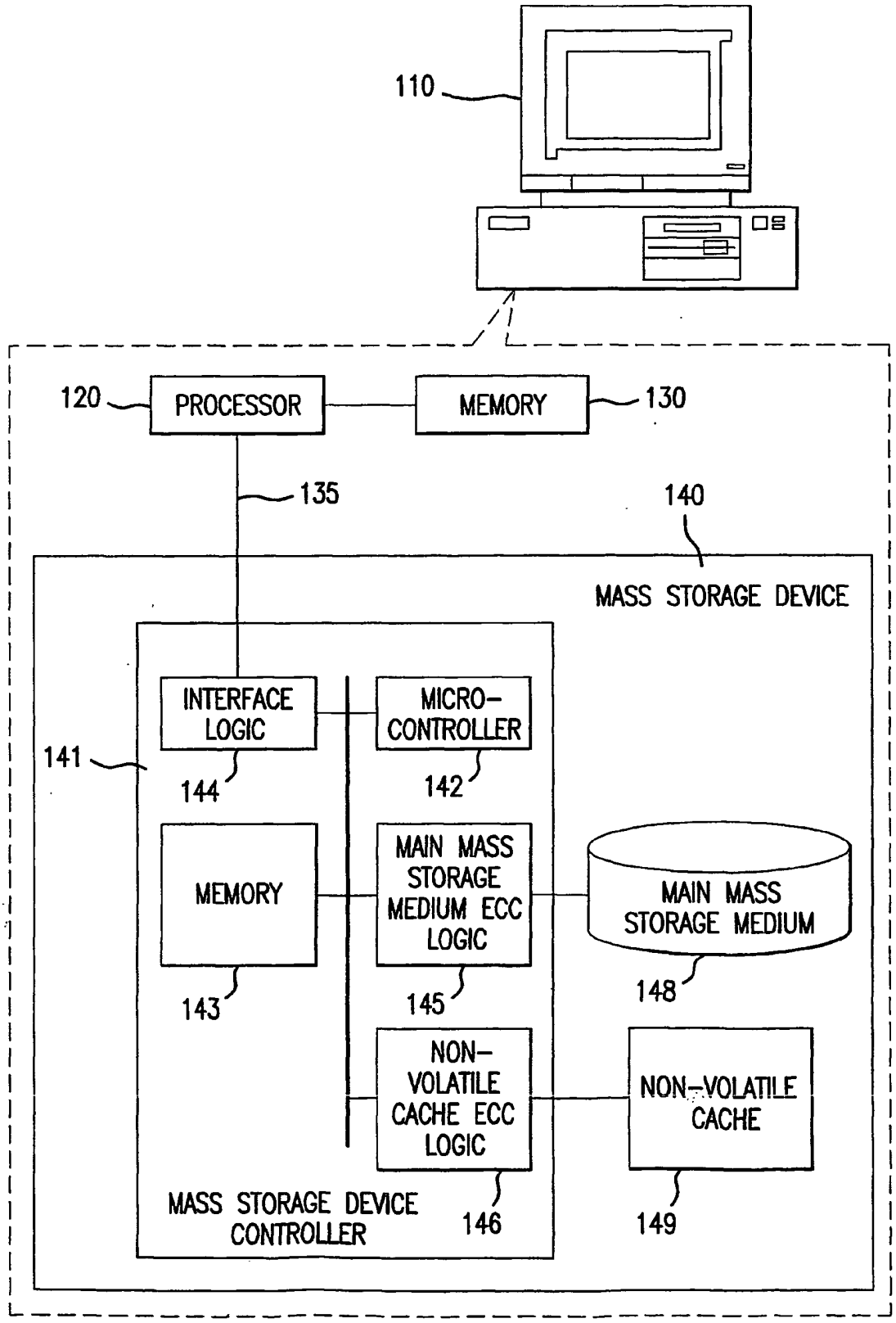

110
120
PROCESSOR
MEMORY
130
135
140
MASS STORAGE DEVICE
INTERFACE LOGIC
MICRO-CONTROLLER
141
144
142
MEMORY
MAIN MASS STORAGE MEDIUM ECC LOGIC
MAIN MASS STORAGE MEDIUM
143
145
148
NON-VOLATILE CACHE ECC LOGIC
NON-VOLATILE CACHE
MASS STORAGE DEVICE CONTROLLER
146
149

FIG.1

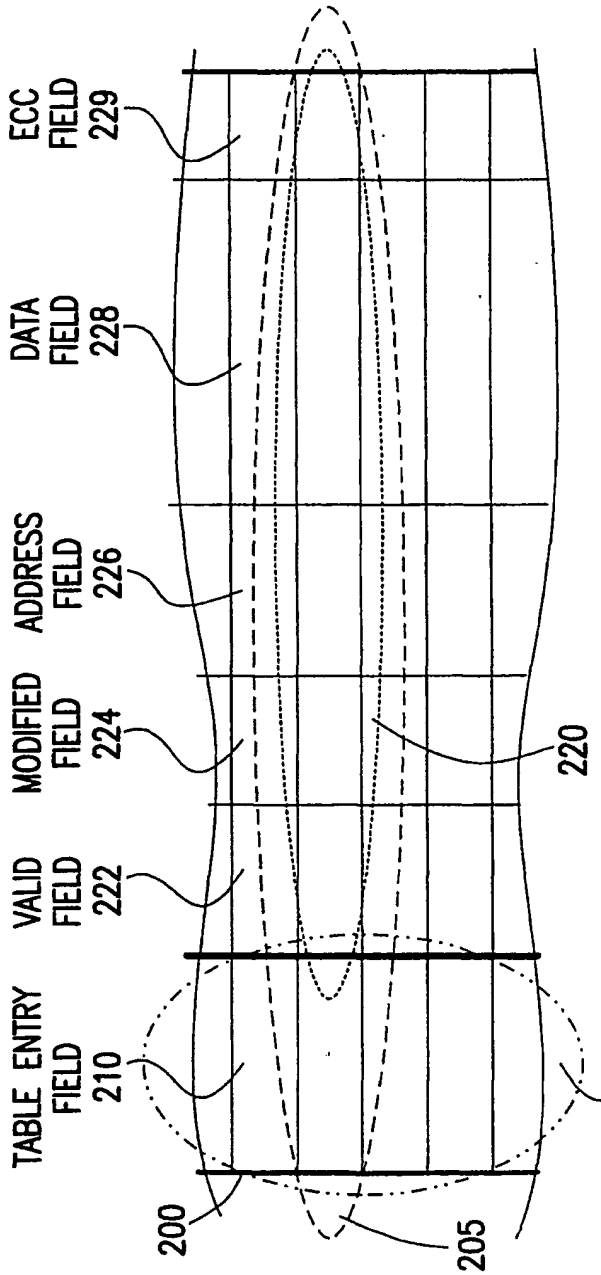

ECC FIELD 229
DATA FIELD 228
ADDRESS FIELD 226
MODIFIED FIELD 224
VALID FIELD 222
TABLE ENTRY FIELD 210
220
215
200
205

FIG.2

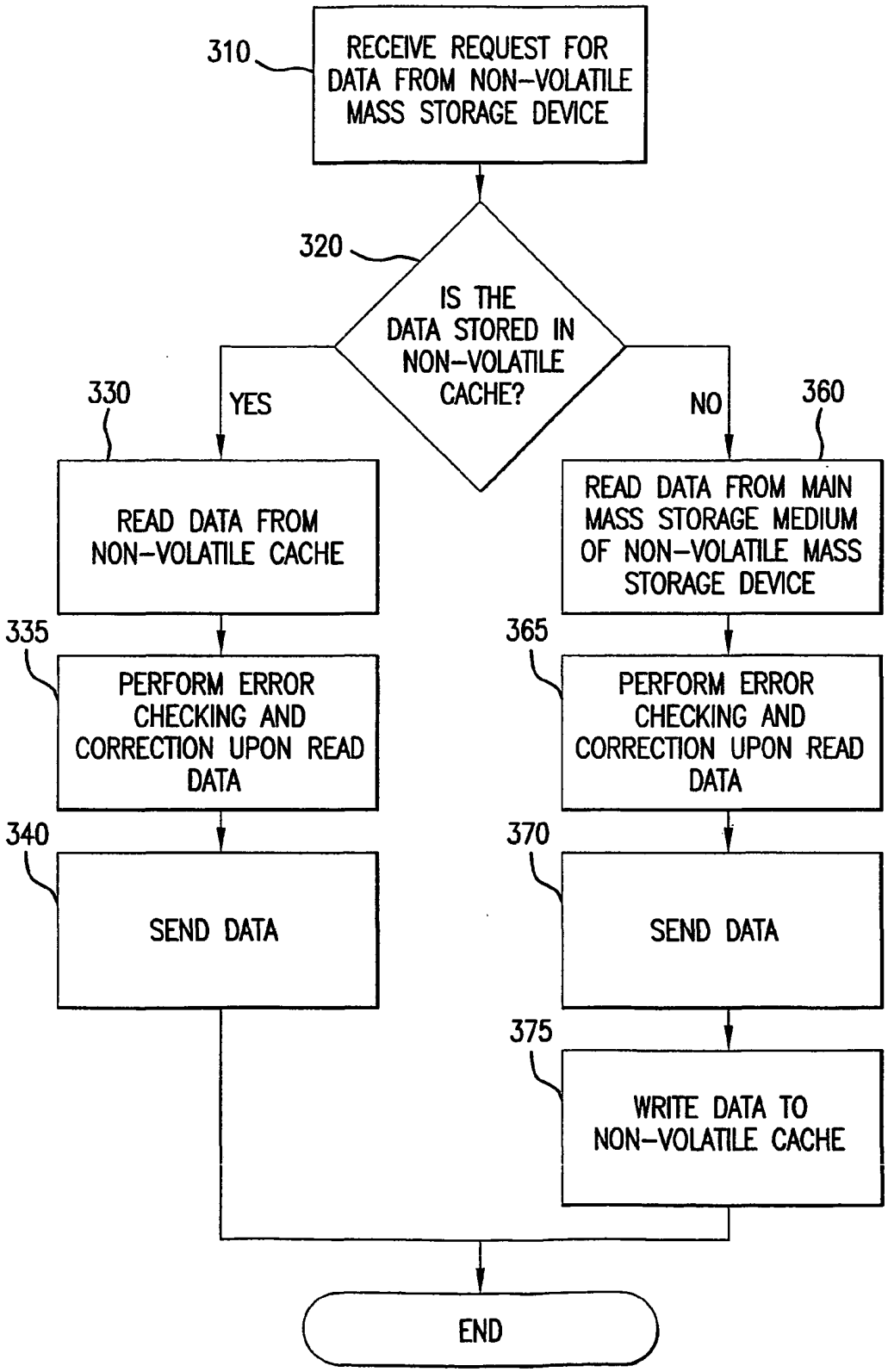

310
RECEIVE REQUEST FOR DATA FROM NON-VOLATILE MASS STORAGE DEVICE
320
IS THE DATA STORED IN NON-VOLATILE CACHE?
YES
NO
330
READ DATA FROM NON-VOLATILE CACHE
335
PERFORM ERROR CHECKING AND CORRECTION UPON READ DATA
340
SEND DATA
360
READ DATA FROM MAIN MASS STORAGE MEDIUM OF NON-VOLATILE MASS STORAGE DEVICE
365
PERFORM ERROR CHECKING AND CORRECTION UPON READ DATA
370
SEND DATA
375
WRITE DATA TO NON-VOLATILE CACHE
END

FIG.3

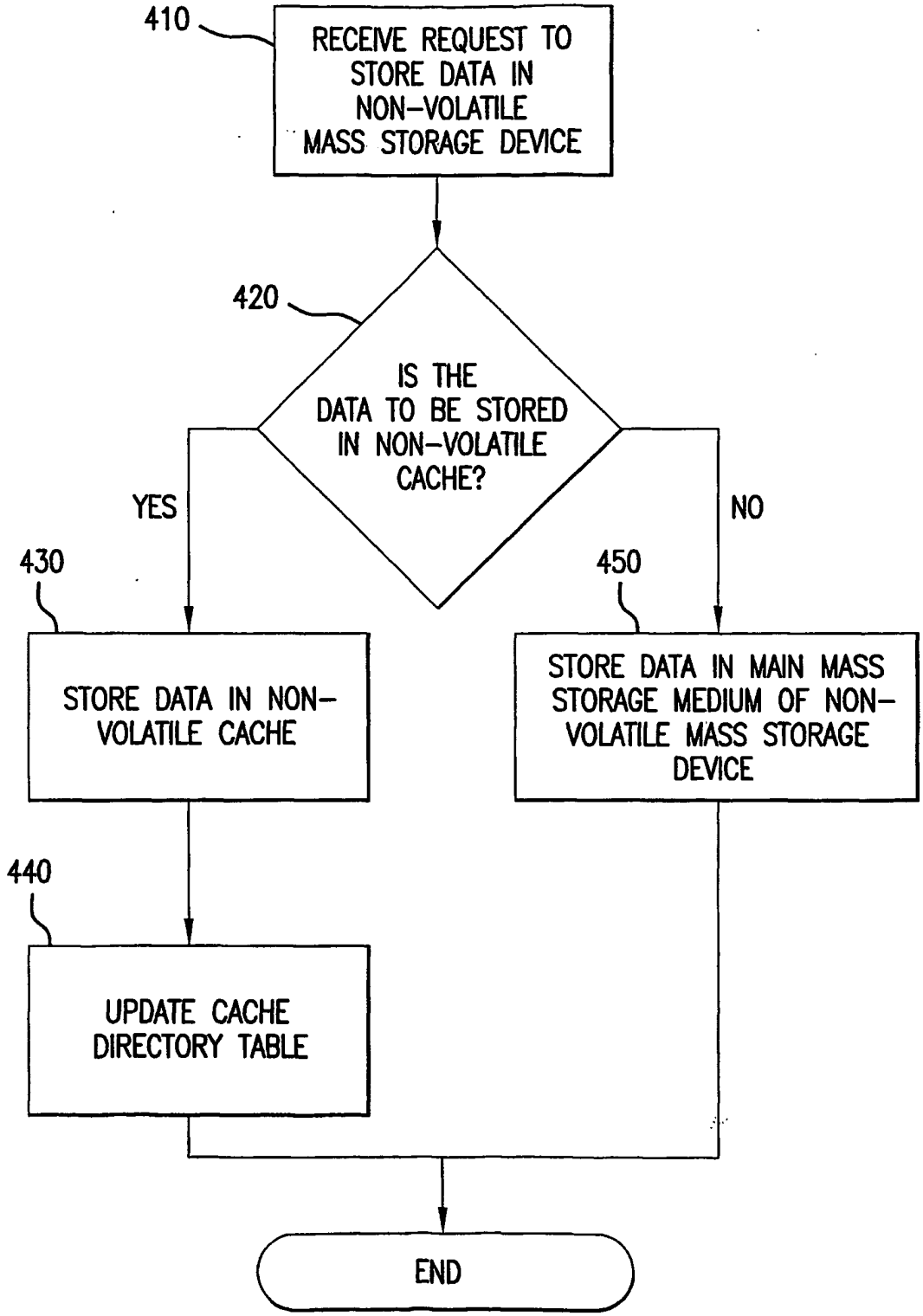

410
RECEIVE REQUEST TO STORE DATA IN NON-VOLATILE MASS STORAGE DEVICE
420
IS THE DATA TO BE STORED IN NON-VOLATILE CACHE?
YES
NO
430
STORE DATA IN NON-VOLATILE CACHE
450
STORE DATA IN MAIN MASS STORAGE MEDIUM OF NON-VOLATILE MASS STORAGE DEVICE
440
UPDATE CACHE DIRECTORY TABLE
END

FIG.4